REMARKS

Applicants thank the Examiner for the thorough consideration given the present application.

Claims 1-12 are pending, of which claims 1, 8, and 11 are independent. Claims 1, 2, and 4-7 are amended to address the informalities noted on page 2 of the Office Action, rendering moot the objection to claims 4-7 and rejection of claims 1 and 2 under 35 U.S.C. §112, second paragraph. Claim 1 now positively defines opening ***and*** closing the sockets. Claim 12 is the same as claim 3, but depends on claim 2 instead of claim 1.

Applicants traverse the rejection of claims 1 and 2 under 35 U.S.C. §103(a) as being unpatentable over the Aline et al. article in view of Chang et al. (U.S. Patent No. 6,134,584), and the rejection of claim 3 as being obvious over Aline et al. in view of Chang et al., and the Fielding et al. article.

While not acquiescing to any rejection, but merely to advance prosecution of the present application, independent claim 1 is amended to recite a method of permitting secure access between a service external to a network firewall and a client internal to the firewall having a combination of steps, including (a) effecting an HTTP GET operation or equivalent thereof from the client to establish a communications socket at the client for communicating data between the service and the client; (b) closing the existing communications socket and opening a new

communications socket at the client for communicating data between the service and the client by performing another GET operation or equivalent thereof after a predetermined time interval; and (c) repeating step (b) while access between the service and the client is required to continue.

Added independent claim 8 recites an apparatus for permitting secure access through a network firewall with a service having a combination of elements, including a communications interface for interfacing the apparatus with the network, the communications interface being arranged to open and close communications sockets; a first control arrangement for using the communications interface to effect a first HTTP GET operation or equivalent thereof in respect of the service thereby to cause the latter to establish a communications socket for communicating data between the service and the client; a second control arrangement for using the communications interface to effect another GET operation or equivalent thereof in respect of the service a predetermined time interval after a most recent GET operation effected by the apparatus in respect of the service, thereby to close the existing the communications socket and to open a new communications socket for communicating data between the service and the client; and a third control arrangement for causing the second control arrangement to terminate its operation when access between the service and the client is no longer required.

Added independent claim 11 is directed to a computer-readable medium storing a computer program arranged to condition a program-controlled networked computer, when executed by the latter, to access a service beyond a network firewall having a combination of steps, including (a) effecting an HTTP GET operation or equivalent thereof from the client to establish a communications socket at the client for communicating data between the service and the client; (b) after a predetermined time interval effecting another GET operation or equivalent thereof to close the existing communications socket and open a new communications socket at the client for communicating data between the same the service and the client; and (c) repeating step (b) while access between the service and the client is required to continue.

None of the cited references, including Aline et al., Chang et al., and Fielding et al., discloses or suggests a method, apparatus, or computer-readable medium having the above-noted features. One of ordinary skill in the art would not have been motivated to combine Aline et al. and Chang et al. to arrive at the steps of claims 1 and 2 or the apparatus of claim 8 or the medium of claim 11.

In claim 1, a first GET operation (or equivalent) is effective to establish a socket on the client for receiving data from a service. After a predetermined interval, another GET operation (or equivalent) closes that socket and opens another

socket in respect of the same service. As is made clear in amended claim 1, step (a) is not repeated. Support for this amendment to claim 1 can be found in the specification as filed, e.g., at page 6, lines 29-36, which corresponds to step (a), and line 44, which (the demon "startPeriodicTimeOutDemon") corresponds to the final step of claim 1, step (b) being the function "switchGETstream" called by the demon and discussed on page 7.

The purpose of the second GET operation is to prevent the connection from being closed unexpectedly by a proxy server that only allows the connection to be open for a time *P*. The "predetermined interval" between GET operations is, therefore, less than the time *P* (see page 5, lines 14-30, of the specification as filed). As recited in amended claim 2, the "predetermined interval" is set with reference to *P*.

Applicants cannot agree with the characterization of Aline et al. set forth in the Office Action. Paragraph 1 on page 2 discusses the actions of the server program. The server program starts by opening a new socket on the server. This is the first line of the second code block on page 2, where a socket object "sock" is created: Socket sock = servSock.accept 0. A connection is then established with the client, and data is exchanged. The server program then closes the socket by invoking the "close" on the socket object "sock" (see the last line of the third code block on page 2): sock.Close 0.

From the foregoing, it is clear that the socket being referred to by the referenced passage of Aline et al. is on the server and is a socket directly under control of the server program. In contrast, Applicants' amended claim 1 is concerned with the opening and closing of a socket at the client (and the subsequent opening of a new socket at the client).

Furthermore, it is clear that the "close" method of Aline et al. is not equivalent to a GET operation, since the latter results in an HTTP request passing from the client to the server, whereas the "close" method is purely internal to the server.

Neither Chang et al. nor Fielding et al. cures the deficiencies of Aline et al. as a primary reference. Chang et al. merely shows closing an Internet connection after a predetermined time. If a transfer operation is still ongoing, the transfer is rescheduled for a later time. At column 7, lines 58-65, Chang discusses the possibility of sending a cancellation message from the "initiating machine" to the "destination machine." However, nowhere does Chang et al. disclose or suggest this as occurring at the time the connection is closed. Chang does not disclose sending a second GET message to close one socket and open another.

In view of the foregoing amendments and remarks, favorable reconsideration and allowance are deemed in order.

To the extent necessary during prosecution, Applicants hereby request any required extension of time not otherwise requested and

hereby authorize the Commissioner to charge any required fees not otherwise provided for, including application processing, extension of time, and extra claims fees, to Deposit Account No. 07-1337.

Respectfully submitted,

LOWE HAUPTMAN GILMAN & BERNER, LLP

By: _____
Allan M. Lowe, Reg. No. 19,641

1700 Diagonal Road, Suite 300
Alexandria, VA 22314
703-684-1111 telephone
703-518-5499 telecopier
AML:rk